The FermiGrid High Availability System

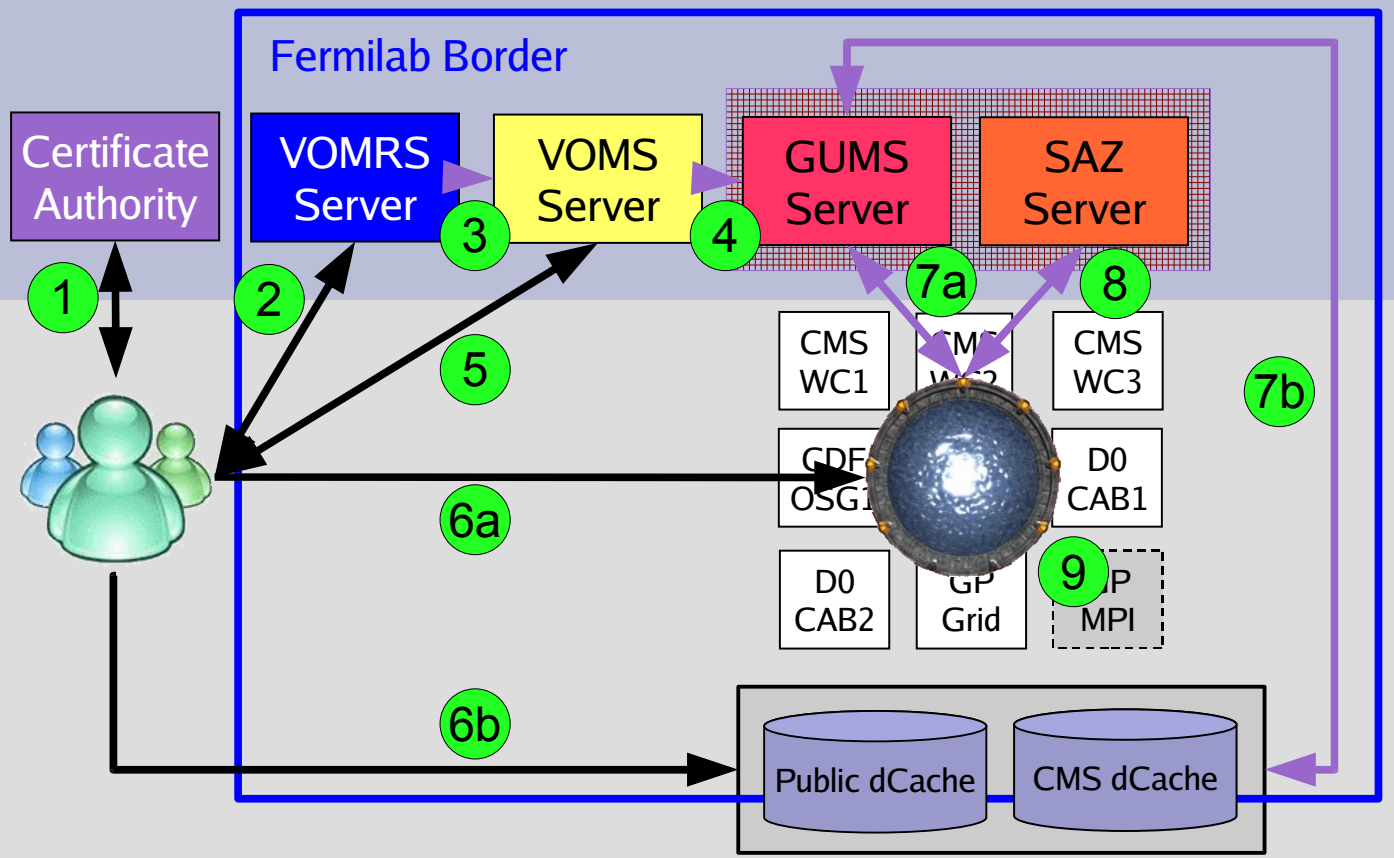
Dan Yocum
Fermilab
February 21, 2008

Introduction

- What is FermiGrid
- Why Does FermiGrid Need to be Highly Available
- How We Made FermiGrid Highly Available
- Testing FermiGrid High Availability
- The Cost of Building FermiGrid High Availability
- Future Work for FermiGrid High Availability
- Conclusion

What is FermiGrid

- Virtual Organization Management Server (VOMS)
  - Attaches extended key attributes to x509 certificates
  - fermilab, dzero, sdss, des, gadu, nanohub, ilc, lqcd, i2u2, osg
- Sitewide Globus Gatekeeper
  - CMSWC1, CMSWC2, CMSWC3, CDFOSG1, CDFOSG2, CDFOSG3, CDFOSG4, D0CABOSG1, D0CABOSG2, GPGRID, et al.
- Grid User Management Server (GUMS)
  - Maps every user to a local UID
    - Compute and storage resources
  - Peak of 1.1M mappings per day
- Site AuthoriZation Server (SAZ)
  - Authorizes grid batch processing jobs
  - Peak of 300K authorizations per day



1 –User obtains x509 certificate (one time only –renew periodically)
2 –User registers cert with VO via VOMRS, requests group, subgroup, and role (one time only –can update)
3 –VOMRS syncs to VOMS (every 30 minutes)
4 –GUMS pulls DN+FQAN from VOMS (every 60 minutes)
5 –User obtains voms proxy certificate (when necessary)
6a –User submits job to FermiGrid gateway
7a –Gatekeeper maps DN+FQAN against GUMS
8 –Gatekeeper authorizes user against SAZ
9 –Gatekeeper forwards job to appropriate grid cluster
alternative:
6b –accesses dCache via SRM
7b –dCache maps DN+FQAN against GUMS via gPlazma

Only accessible within fnal.gov

Why does FermiGrid need to be Highly Available

- Without VOMS extended attributes VO members can't map to local UID
- All compute jobs on all worker nodes are mapped by GUMS
- All compute jobs on all worker nodes are authorized by SAZ
- All file access requests to CMS and public dCache systems are mapped by GUMS
- Average of 10,000 mappings **per hour!**
- **These services are critical to data access and data analysis!**

How We Made FermiGrid Highly Available

- Make services fault tolerant
  - Replicate data and services, add failover capabilities (aka active-passive)
  - Vulnerable to overloading, service time-outs
- Make services load balanced
  - Replicate data and services, add load balancer (aka active-active)
  - Solves the (N-1) system failure problem
  - Solves the overloaded system problem, scales linearly with number of systems

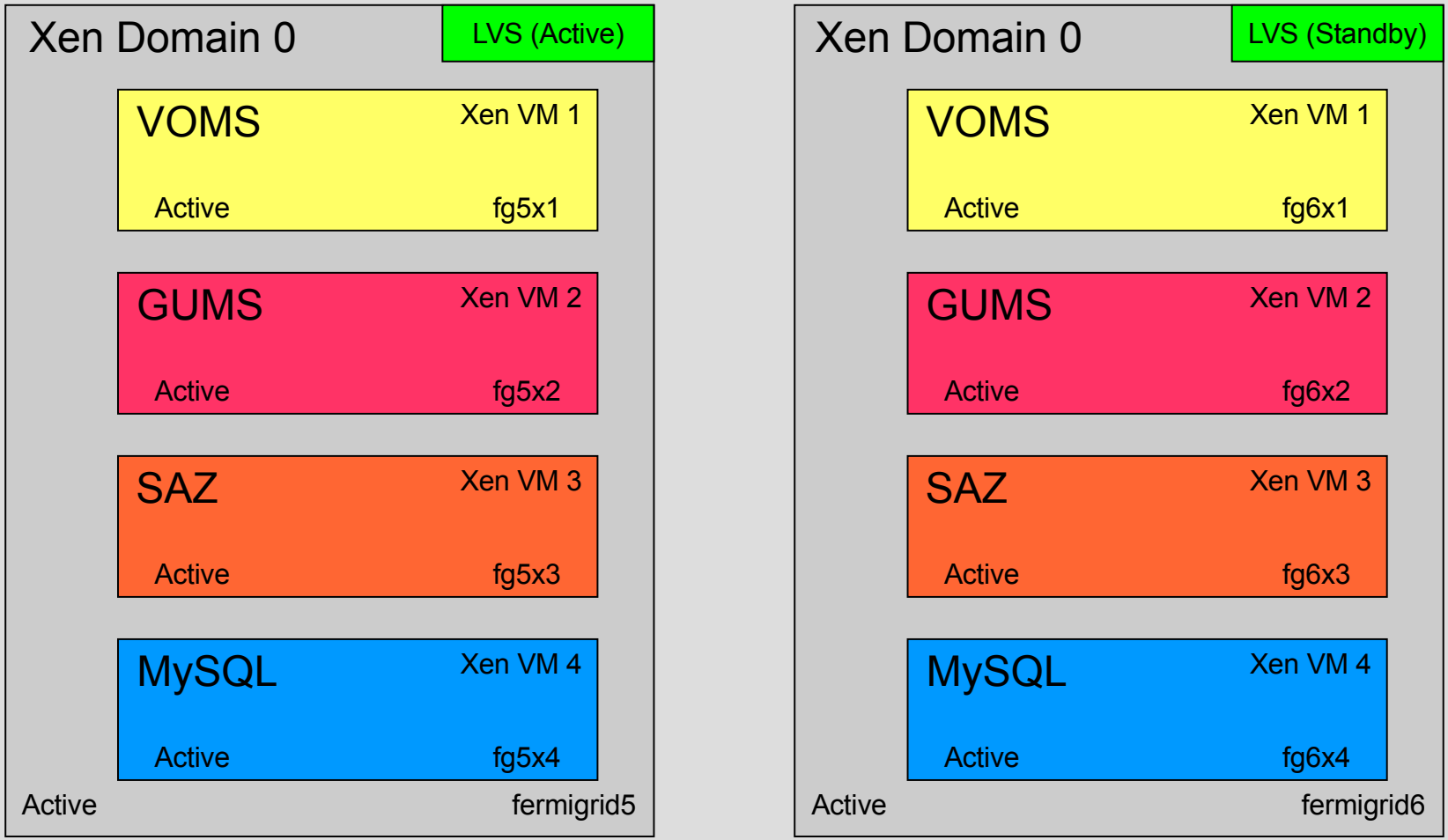
How We Made FermiGrid Highly Available

- More specifically
  - Multiple systems, currently 2 physical machines
    - Hardware details later...
  - Multiple Virtual Machines on each physical machine, currently 4 (tune in Feb 28, 2008)
  - Replicate data across systems
    - Shared-nothing: no shared FS, no FC, no NFS
  - Replicate services on multiple machines
    - Without modifying source code
  - Add HA abstraction layer
    - Single point of contact presented to clients
      - voms.fnal.gov, gums.fnal.gov, saz.fnal.gov, fg-mysql.fnal.gov
    - Multiple servers on the backend
      - fg5x1/fg6x1, fg5x2/fg6x2, fg5x3/fg6x3, fg5x4/fg6x4

How We Made FermiGrid Highly Available

- Currently 2 physical machines
  - Dell 2950
  - Dual core, dual CPU, 3GHz
  - 16GB RAM
  - Dual Gigabit ethernet NICs
  - 150GB RAID 1
  - Redundant Power Supplies
- Xen – covered Feb 28 by Steve Timm
  - fermigrid5 hosts fg5x1, fg5x2, fg5x3, fg5x4
  - fermigrid6 hosts fg6x1, fg6x2, fg6x3, fg6x4

How We Made FermiGrid Highly Available



How We Made FermiGrid Highly Available

- MySQL Replication
  - Investigated several technologies
    - MySQL Cluster
    - drbd – distributed remote block device
    - mysqlhotcopy
    - mysqldump + rsync
    - Multi-master replication (requires MySQL >v5.0.2)
  - Decided on Multi-master replication
  - Scales well to 10 systems
    - currently only using 2
  - Near real-time replication, 1.1ms for 1KB record
  - Database server outages handled correctly
    - Logs replayed after server rejoins chain

How We Made FermiGrid Highly Available

- MySQL Replication, cont'd
  - Complete failure requires full copy (obviously)
  - >2 systems require special techniques to close the circular chain in the event of failure
- Service Replication
  - 2 VOMS “servers”, one each on fg5x1 & fg6x1
    - Each machine serves 12 VOs
  - 2 GUMS servers, one each on fg5x2 & fg6x2
  - 2 SAZ servers, one each on fg5x3 & fg6x3
  - All services use replicated MySQL servers on fg5x4 and fg6x4
  - No source code modifications

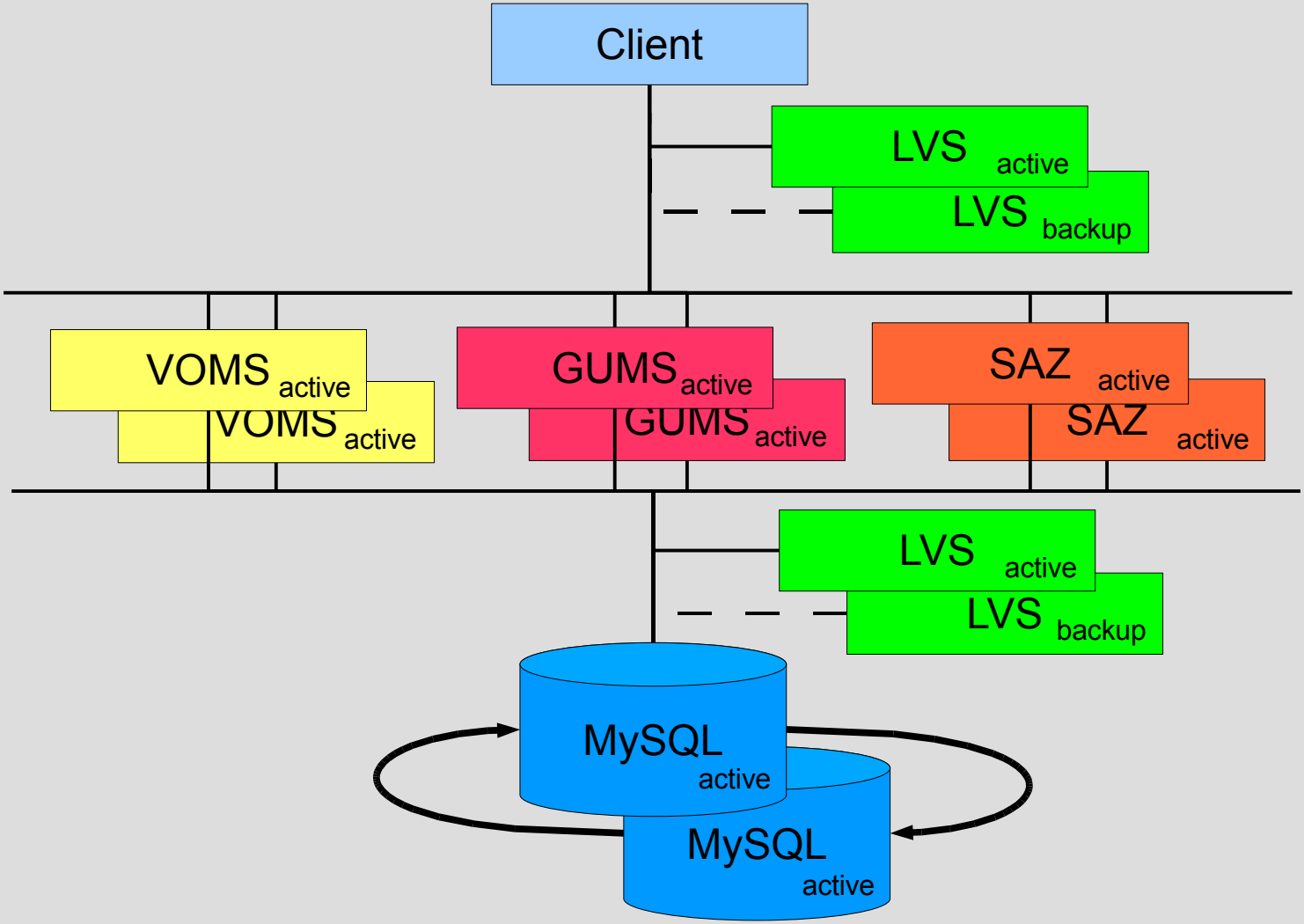
How We Made FermiGrid Highly Available

- High Availability and Load Balancing
  - Investigated several technologies
    - Heartbeat (active-passive)
    - Round-robin DNS (not supported at Fermi, $$)
    - Linux Virtual Server (LVS)
  - Selected Linux Virtual Server
    - Piranha product from Red Hat
    - Use Direct Routing (LVS-DR) method
  - Listens on voms.fnal.gov, gums.fnal.gov, saz.fnal.gov, fg-mysql.fnal.gov
  - Re-directs connections based on IP+port to backend “real servers”
    - Uses weighted least connections (WLC) to schedule connection requests
      - Other algorithms available

How We Made FermiGrid Highly Available

- High Availability and Load Balancing, cont'd
  - In LVS-DR, real servers respond directly to client
  - LVS “pings” services for availability
    - Removes service from scheduling if unavailable
    - Adds service back in when available
  - LVS master server is fermigrid5
  - LVS backup server is fermigrid6
    - Active-passive configuration
    - Failover in 6 seconds
    - Active connections to backend servers are maintained
  - If a service or real server fails during an open connection, the connection is lost.

How We Made FermiGrid Highly Available



Note 1: All network connections are on the public network
Note 2: LVS directors displayed separately for convenience – they are the same in reality

Testing FermiGrid High Availability

- Each service tested for load, stability, fault tolerance before deploying in production
  - 25 clients
    - GUMS successfully mapped at 112Hz, 9.7M/day
      - System load – 9.2, CPU – 92%
    - SAZ successfully authorized at 12Hz (1.1M/day)
      - System load – 12, CPU – 100%
    - VOMS works
      - No stress testing performed – only 1,800 requests/day
    - MySQL successfully queried at 125Hz (10.8M/day)
      - System load – 0.2, CPU – 10%
    - Each service performed without fault for many hours under these extreme loads

Testing FermiGrid High Availability

- Fault tolerance testing was successful
  - Simulated failures of VOMS, GUMS, and SAZ
    - Disabled network
      - New requests not routed to failed server
    - Re-enabled network
      - server added back to the pool for scheduling
    - Open connections during service failure are lost.
      - No cheap and easy way to migrate TCP syn/ack sequences to another real server.
      - Affected number of connections is very small (1-2)
  - Simulated failure of MySQL server
    - After re-enabling server, transaction logs replayed automatically
  - Simulated LVS failure
    - Hot standby LVS server successfully took over in 6 seconds
    - Open connections maintained across failover

The Cost of Building FermiGrid High Availabilty

- Hardware
  - 2 Dell 2950s - $18,975.00
- Software
  - Open Source Software - $0
- Manpower
  - Approximately 50% FTE

The Cost of Building FermiGrid High Availabilty

- Timetable
  - Equipment received on Jun 5, 2007
  - OS install and Xen work started early Jul, 2007
  - Load balancing work started on Aug 23, 2007
  - MySQL replication completed Aug 24, 2007
  - VOMS install completed Sep 27, 2007
  - GUMS install completed Sept 28, 2007
  - SAZ and LVS installs completed mid Oct, 2007
  - All testing completed Nov 2, 2007
  - All services placed into production Dec 3, 2007
  - Documentation written in parallel with work performed

Future Work for FermiGrid High Availability

- Making the following services highly available are dependent on furlough schedules and available funds
  - Squid
  - Ganglia
  - syslog-ng
  - Myproxy
  - Condor
  - Globus gatekeeper

Conclusion

- FermiGrid is the site job submission portal
  - Provides authentication and authorizations for all compute jobs and dCache access
- FermiGrid must be available at all times
  - 5,000 – 50,000 “jobs” per hour depend on FermiGrid services being available
- MySQL multi-master replication and Linux Virtual Server technology provides a stable, fault tolerant and highly available system

Conclusion

- Testing we have performed shows FermiGrid HA system is stable and capable of providing 10x our current needs
  - Scaling up only requires more hardware
- Cost of LVS is much, much less than a Cisco Round-robin DNS blade.

Bonus Material

- More LVS details
  - 3 modes of operation
    - Network address translation (NAT)
      - Real Servers live on a private LAN
      - All traffic flows through LVS director node - bottleneck!
    - Tunneling using IP encapsulation
      - Real Servers live on any public network, LAN or WAN
      - Good for geographically separated Real Servers
      - Real Servers answer directly to client – no bottleneck
      - Real Servers must be able to un-encapsulate IP packets
    - Direct Routing
      - Real Servers and Director **must** live on same physical subnet
      - Director re-writes MAC in address frame and transmits on LAN
      - Real Servers respond directly to client – no bottleneck

Bonus Material

- More LVS details
  - Director maintains a hash table of open connections
    - Hash table is mirrored to backup LVS director
  - For long lived, transactional, and SSL connections, use persistence
  - Solving the ARP problem
    - Option A: use Transparent Proxy on Real Servers
      - Virtual Service IP (VIP) is not actually running on Real Server
      - Configured using iptables
    - Option B: use a real IP on a hidden interface (lo:0)
      - Enable net.ipv4.conf.all.arp_ignore to not respond to arp requests on VIP interface

Bonus Material

- Multi-homed SSL web servers on the same physical system
  - Can't use Transparent Proxy – must use IP on hidden interface
    - For >1 web server with separate hostnames running on the same machine, apache must bind to the IP of the service running (i.e., Listen 1.2.3.4)
      - voms.opensciencegrid.org
      - voms.fnal.gov
    - Separate VirtualHost directives containing the correct x509 service certificate with the appropriate hostname in the CN must be specified for each IP apache listens on.

Bonus Material

- Issues with multi-master MySQL replication
  - External data (i.e., on the filesystem) referenced within the MySQL database are not replicated
    - Use rsync
  - Replication across timezones, national languages, etc. can cause problems
    - All our servers are local
  - Auto incrementing column problem
    - Use auto_increment_increment (10)
    - Use auto_increment_offset (1,2,3,...)